Supplementary Information

Anubis: Bayesian optimization with unknown feasibility constraints for scientific experimentation

Riley Hickman,[1,2,3,*] Gary Tom,[1,2,3] Yunheng Zou,[2,3]
Matteo Aldeghi,[1,2,3,†] and Alán Aspuru-Guzik[1,2,3,4,5,6,7,‡]

[1] *Chemical Physics Theory Group, Department of Chemistry, University of Toronto, Toronto, ON M5S 3H6, Canada*
[2] *Department of Computer Science, University of Toronto, Toronto, ON M5S 3G4, Canada*
[3] *Vector Institute for Artificial Intelligence, Toronto, ON M5S 1M1, Canada*
[4] *Acceleration Consortium, University of Toronto, Toronto, ON M5S 3E5, Canada*
[5] *Department of Chemical Engineering & Applied Chemistry, University of Toronto, Toronto, ON M5S 3E5, Canada*
[6] *Department of Materials Science & Engineering, University of Toronto, Toronto, ON M5S 3E4, Canada*
[7] *Lebovic Fellow, Canadian Institute for Advanced Research, Toronto, ON M5G 1Z8, Canada*
[*] *riley.hickman@mail.utoronto.ca*
[†] *Current address: Bayer Research and Innovation Center, 238 Main St, Cambridge, MA 02142, USA*
[‡] *alan@aspuru.com*

S.1. SUPPLEMENTARY INFORMATION

A. Effect of the acquisition function on optimization performance

1. Acquisition functions

Two acquisition functions are considered in this work. The *expected improvement* (EI) acquisition function is a commonly used function in the Bayesian optimization literature. Given a black-box function f to be optimized, let f' be the best value of this function observed so far. EI proposes to evaluate f at the parameter point at which we expect to improve on f' the most. The utility of a parameter point $\boldsymbol{x}$ is then $\max(0, f' - f(\boldsymbol{x}))$, and the EI acquisition function is the expectation value of this utility,

$$\alpha(\boldsymbol{x}) = (\mu(\boldsymbol{x}) - f' - \xi)\,\Phi(Z) + \sigma(\boldsymbol{x})\phi(Z)\ , \tag{10}$$

$$Z = \frac{\mu(\boldsymbol{x}) - f' - \xi}{\sigma(\boldsymbol{x})}\,. \tag{11}$$

where $\mu(\boldsymbol{x})$ and $\sigma(\boldsymbol{x})$ are respectively the surrogate model's mean and standard deviation. Φ and ϕ are the CDF and PDF of the standard normal distribution, respectively. ξ is a tradeoff parameter which determines the amount of exploration during optimization. We set $\xi = 0.01$ in all experiments.

The *upper confidence bound* (UCB) acquisition function is

$$\alpha(\boldsymbol{x}) = \mu(\boldsymbol{x}) + \sqrt{\beta}\sigma(\boldsymbol{x})\,, \tag{12}$$

where $\mu(\boldsymbol{x})$ and $\sigma(\boldsymbol{x})$ are respectively the surrogate model's Gaussian mean and standard deviation. $\beta > 0$ is a tradeoff parameter that weights the importance of variance based sampling in the acquisition function. We set $\beta = 0.2$ in all experiments.

We perform a series of experiments to determine the effect of acquisition function choice on unconstrained and constrained optimization analytic surfaces. The 2d continuous *Dejong* surface and the 2d categorical *Michalewicz* surface are used. The unconstrained and constrained versions of the surface are used, and the *FWA* and *FCA-0.2* strategies are tested for the constrained tests. Results from these comparative experiments are shown in Fig. S1. In all tests, the UCB acquisition outperformed or had equivalent performance compared to the EI acquisition. Thus, we elected to use UCB in all experiments reported in this work.

2. Acquisition function optimization

Atlas contains 3 acquisition function optimization strategies: (i) a constrained gradient optimizer based on SLSQP[1] from the BoTorch library,[2] (ii) a constrained genetic algorithm (GA) optimizer based on the PyMOO

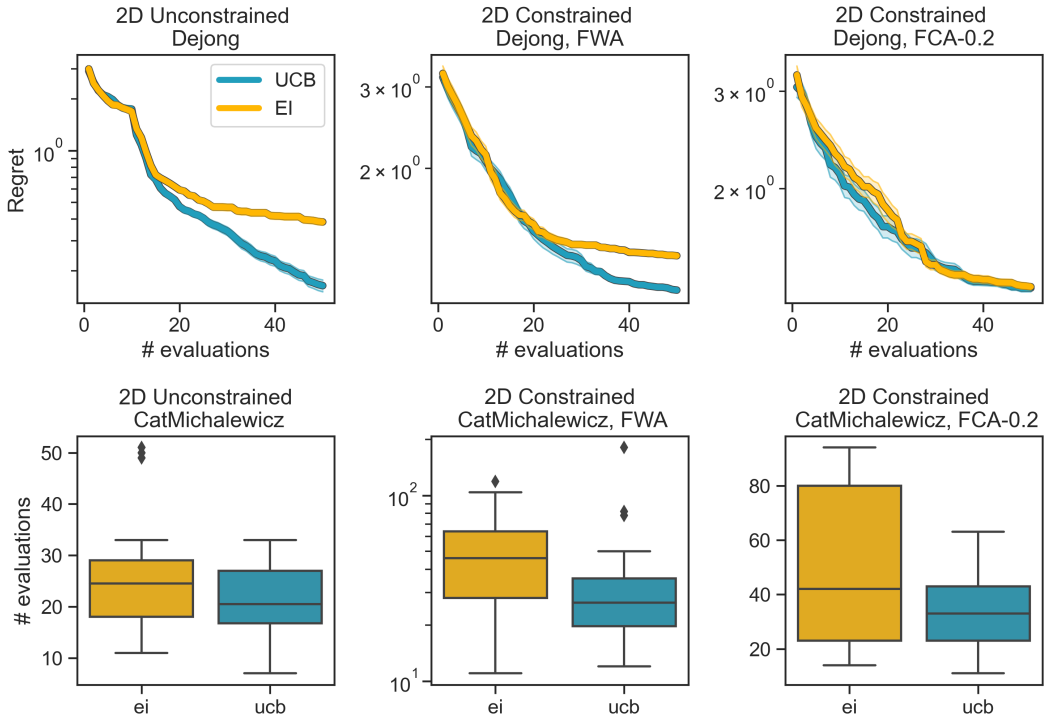

FIG. S1. Comparison of the EI and UCB acquisition functions on 2d unconstrained and constrained analytic surfaces. The top row shows regret traces on the continuous 2d *Dejong* surface. The bottom row shows the number of surface evaluations needed to identify the optimum of the 2d categorical *Michalewicz* surface. For each experiment, strategies were run independently 40 times.

library,[3] and (iii) a constrained GA based on the DEAP library.[4,5] Importantly, all strategies are compatible with optimization over parameter domains with *a priori* known constraints.[6] This is essential for the *FCA* feasibility-aware acquisition strategy, where at each iteration $\alpha(\boldsymbol{x})$ must be optimized according to a learned constraint oracle $P\,(\text{feasible}|\boldsymbol{x}) > t$, where $t \in [0, 1)$.

For fully-continuous parameter problems, the gradient-based acquisition optimization makes use of the BoTorch library, which uses the SLSQP algorithm. For highly constrained problems (*i.e.* those for which a significant fraction of the parameter space is infeasible), we found that the SLSQP optimizer undergoes serious convergence issues. Ultimately, this significantly increases the overall per-iteration runtime of the optimization procedure. With our in-house approach for constrained acquisition optimization using DEAP, we are able to preserve optimization performance while significantly reducing the per-iteration runtime. Empirical comparisons of the two methods are shown in Fig. S.1 A 2 and in Table S.1 A 2. We use the 2d (un)constrained StylinskiTang surface, testing the *FCA* strategy with $t = \{0.2, 0.5, 0.8\}$ for the constrained case. While the genetic acquisition optimizer is slightly slower in the unconstrained case, constrained acquisition optimization only requires ≈ 10 seconds per iteration, while the gradient optimizer can take as long as 250 seconds per iteration. Thus, we elect to use the DEAP acquisition optimization strategy for all experiments in this paper.

	Acquisition optimization time (s)	
Setting	**Gradient**	**Genetic**
Unconstrained	0.207 ± 0.002	2.219 ± 0.001
Constrained (FCA, $t = 0.2$)	71.28 ± 3.36	11.03 ± 0.02
Constrained (FCA, $t = 0.5$)	253.94 ± 11.28	9.12 ± 0.26
Constrained (FCA, $t = 0.8$)	60.84 ± 3.43	9.88 ± 0.25

TABLE S1. Average time taken for gradient and genetic strategies to optimize acquisition functions. Reported values are averaged over 100 iterations and 40 independent runs. The 2d (constrained) StyblinskiTang analytical function is used. We do not find that the number of transpired evaluations to have a significant influence on acquisition optimization time in these experiments.

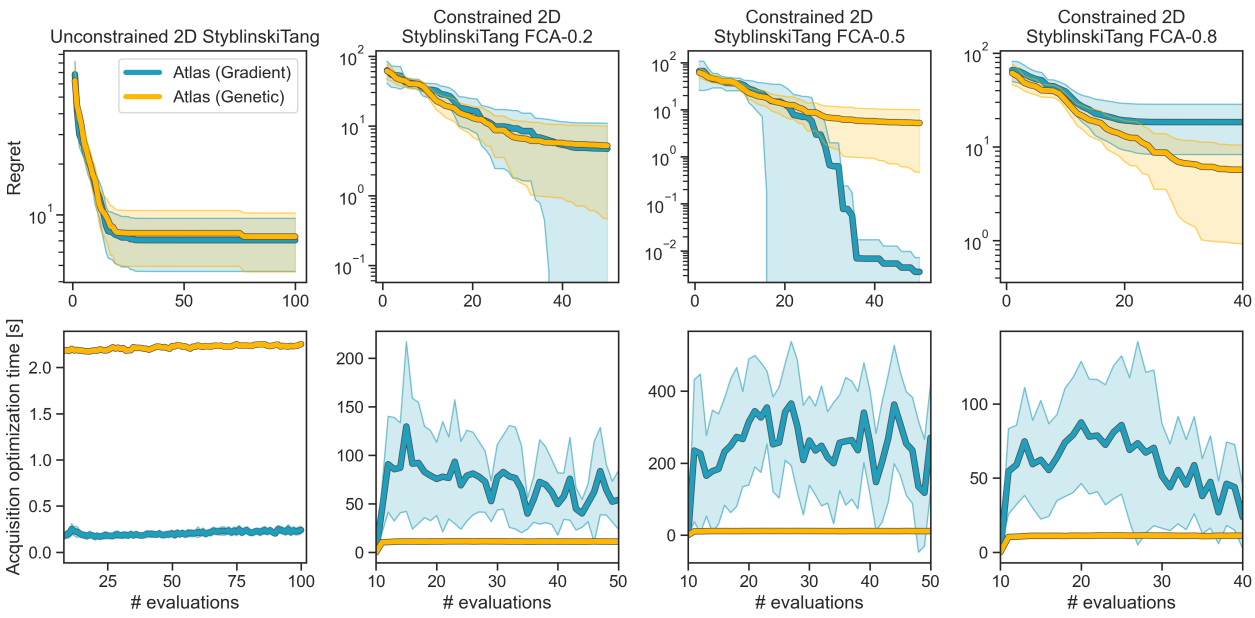

FIG. S2. Comparison of the SLSQP (gradient) and DEAP (genetic algorithm) acquisition optimization strategies in terms of optimization performance and time taken to optimize $\alpha(\boldsymbol{x})$ as a function of transpired evaluations. The unconstrained and constrained 2D StyblinskiTang surface is used in all experiments, using the FCA strategy with $t = \{0.2, 0.5, 0.8\}$ for the constrained case. Each approach was run 40 independent times.

3. Modification to the FIA and FWA feasibility-aware acquisition functions

As addressed in main text Sec. III.B, we report a minimum-filtered version of the *FWA* and *FIA* functions. To the best of our knowledge, this is a novel modification. The output of the GP feasibility classifier, $P\left(\text{feasible}|\boldsymbol{x}\right)$, is replaced in main text Equations 5 and 7 by

$$\rho(\boldsymbol{x}) = \min\left[0.5, P\left(\text{feasible}|\boldsymbol{x}\right)\right] . \tag{13}$$

Effectively, this modification biases the search away *only* from areas where the classifier believes are infeasible. Put another way, all parameter space regions where $P\left(\text{feasible}|\boldsymbol{x}\right) \geq 0.5$ are treated as equally promising. We found that without using $\rho(\boldsymbol{x})$, the search would often become trapped in parameter regions where $P\left(\text{feasible}|\boldsymbol{x}\right) \approx 1$, *i.e.* $P\left(\text{feasible}|\boldsymbol{x}\right)$ would overpower the $\alpha(\boldsymbol{x})$ containing factor and term in main text Equations 5 and 7, respectively. Validation of this claim is shown in Fig. S3, where we compare optimization performance of the minimum filtered and original versions of *FIA* and *FWA* functions using the 2d constrained HyperEllipsoid and Branin-Hoo surfaces. In each case, the minimum filtered version achieves a significantly lower regret value after 50 evaluations, while incurring a comparable number of infeasible measurements.

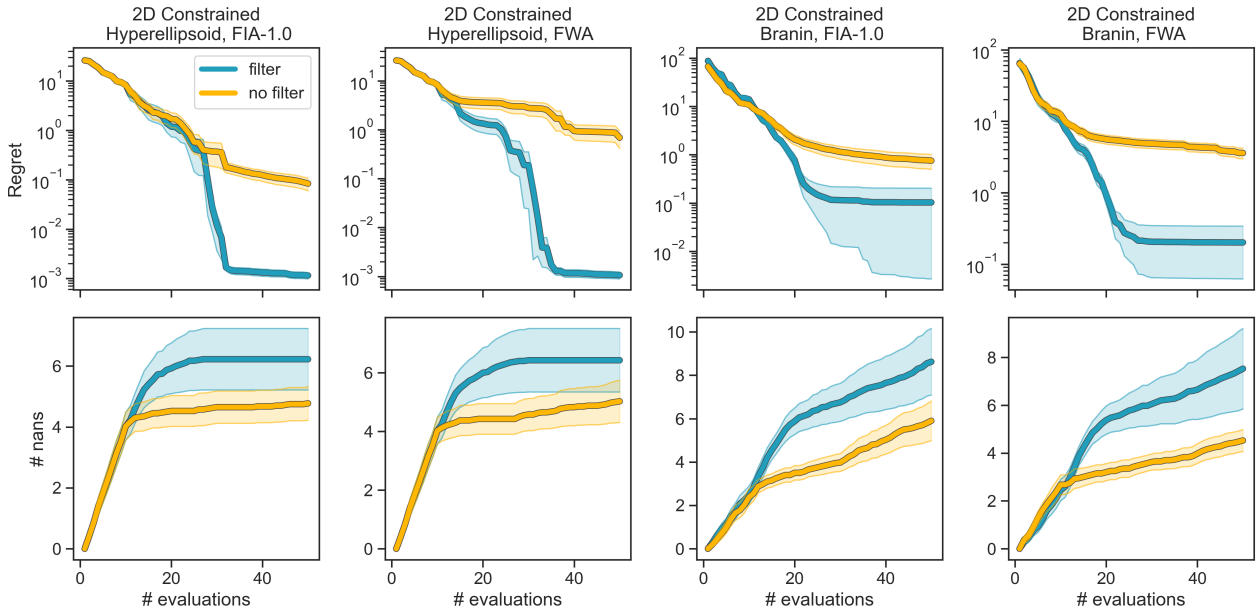

FIG. S3. Comparison of *FIA* and *FWA* feasibility-aware optimization strategies' performance on the 2d constrained Hyper-Ellipsoid and Branin-Hoo surfaces. "Filtered" strategies have $P\,(\text{feasible}|\boldsymbol{x})$ replaced by $\rho(\boldsymbol{x}) = \min\left[0.5, P\,(\text{feasible}|\boldsymbol{x})\right]$. The DEAP acquisition optimizer is used in all tests. Each approach was run 40 independent times.

B. Analytical tests on discrete surfaces

	CatSlopeConstr		CatDejongConstr		CatMichalewiczConstr		CatCamelConstr	
Strategy	% explore (↓)	% infeas (↓)	% explore (↓)	% infeas (↓)	% explore (↓)	% infeas (↓)	% explore (↓)	% infeas (↓)
Random								
Naïve-replace	22.2 ± 1.8	22.4 ± 0.7	82.5 ± 2.8	15.9 ± 0.5	9.1 ± 0.4	18.1 ± 0.6	44.1 ± 2.2	$\mathbf{21.8 \pm 0.4}$
Naïve-ignore	$\mathbf{3.7 \pm 0.1}$	32.3 ± 1.1	$\mathbf{3.3 \pm 0.1}$	26.8 ± 1.0	10.1 ± 0.5	42.4 ± 1.4	13.9 ± 0.5	$\mathbf{22.2 \pm 0.5}$
Naïve-surrogate	4.6 ± 0.2	33.5 ± 1.0	5.8 ± 0.4	31.2 ± 1.0	$\mathbf{7.6 \pm 0.3}$	24.7 ± 0.8	20.9 ± 0.9	$\mathbf{21.4 \pm 0.4}$
FWA	$\mathbf{3.6 \pm 0.1}$	28.3 ± 1.0	10.5 ± 2.1	26.4 ± 1.3	9.7 ± 0.5	34.8 ± 1.1	$\mathbf{12.3 \pm 0.5}$	$\mathbf{22.7 \pm 0.5}$
FCA-0.2	$\mathbf{3.7 \pm 0.1}$	31.8 ± 1.1	48.5 ± 7.9	21.1 ± 2.1	12.0 ± 0.9	35.7 ± 1.0	$\mathbf{12.3 \pm 0.6}$	$\mathbf{21.5 \pm 0.7}$
FCA-0.5	5.7 ± 0.7	26.8 ± 0.9	76.4 ± 8.5	17.0 ± 0.7	16.9 ± 1.7	22.6 ± 0.9	$\mathbf{13.2 \pm 0.5}$	$\mathbf{22.3 \pm 0.5}$
FCA-0.8	18.8 ± 1.7	$\mathbf{17.5 \pm 0.8}$	43.8 ± 4.5	$\mathbf{10.2 \pm 1.7}$	32.6 ± 4.9	$\mathbf{11.8 \pm 1.4}$	16.9 ± 1.2	$\mathbf{22.3 \pm 0.6}$
FIA-0.5	$\mathbf{3.6 \pm 0.1}$	30.0 ± 0.8	12.0 ± 1.6	26.6 ± 1.1	9.5 ± 0.6	30.9 ± 0.8	$\mathbf{13.0 \pm 0.5}$	$\mathbf{22.3 \pm 0.5}$
FIA-1	$\mathbf{3.7 \pm 0.1}$	30.9 ± 1.0	6.3 ± 0.7	27.0 ± 1.0	$\mathbf{6.8 \pm 0.4}$	27.0 ± 1.7	$\mathbf{12.9 \pm 0.5}$	$\mathbf{22.0 \pm 0.5}$
FIA-2	$\mathbf{3.6 \pm 0.1}$	31.5 ± 1.1	4.4 ± 0.2	30.2 ± 1.1	9.5 ± 0.5	37.4 ± 1.3	13.8 ± 0.4	$\mathbf{22.4 \pm 0.5}$

TABLE S2. Numerical results of benchmark of feasibility-aware acquisition strategies on 2d categorical constrained surfaces. The % explored metric is the percentage of the optimization domain (442 total options) needed for each strategy to identify the single best choice. The % infeas metric is the percentage of all measurements that are infeasible. All strategies are independently run 100 times. The best performing strategies for each application and performance metric are bolded. Statistical hypothesis testing is conducted using Welch's t-test.

Results of the discrete, ordered benchmarks are shown in Fig. S4 and numerical values are summarized in Tab. S4. Results of the categorical, unordered benchmarks are shown in Fig. S5. Here, each run is terminated after identification of the surface's global optimum, and 100 independently seeded runs are executed. The major takeaways from the discrete, ordered benchmark tests are listed below.

- As discussed in main text Sec. IV.A, the *naïve-ignore* function gets stuck repeatedly recommending the same continuous parameters in the absence of a self-avoidance routine. However, for fully discrete/categorical problems, duplicate avoidance is baked into Atlas, and this behavior is no longer possible. In fact, the *naïve-ignore* function is among the most efficient strategies for the discrete *CatSlopeConstr* and *CatDejongConstr* surfaces.

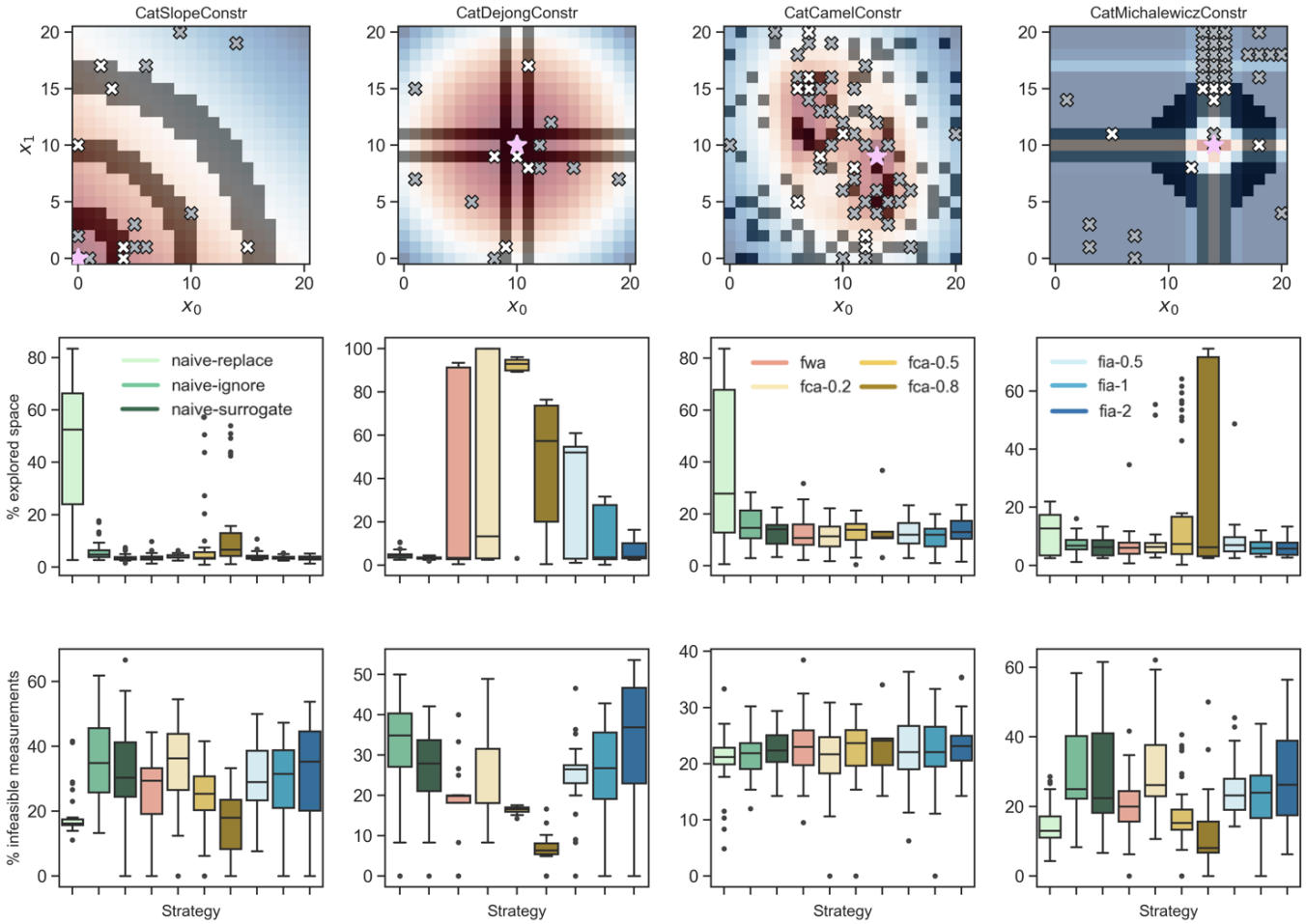

FIG. S4. Constrained optimization benchmarks on analytical functions with discrete, ordered parameters. The upper row shows heatmaps of the discrete optimization domain. Gray crosses show feasible measurements, while white crosses show infeasible measurements, and pink stars show the location of the unconstrained global optima. The middle row shows box-and-whisker plots of the distribution of the percentage of explored parameter space needed for each strategy to identify the global optimum. Each strategy was run 100 independent times. The bottom row shows box-and-whisker plots of the distribution of the number of infeasible measurements queried by each strategy.

- For the *CatCamelConstr* problem, $c(\boldsymbol{x})$ is constructed by randomly constraining a set of discrete options. Therefore, we expect its structure not to be learnable by the classifier. Indeed, the % infeas metric for this problem is statistically equivalent across all strategies in Tab. S4.
- The *FCA-0.8* strategy measures the lowest % infeasible parameters for all discrete surfaces. This, combined with similar outcomes on continuous surface benchmarks and real-world applications, suggest that *FCA* with t set close to 0.8 is the most effective strategy for limiting the constraint violation rate.

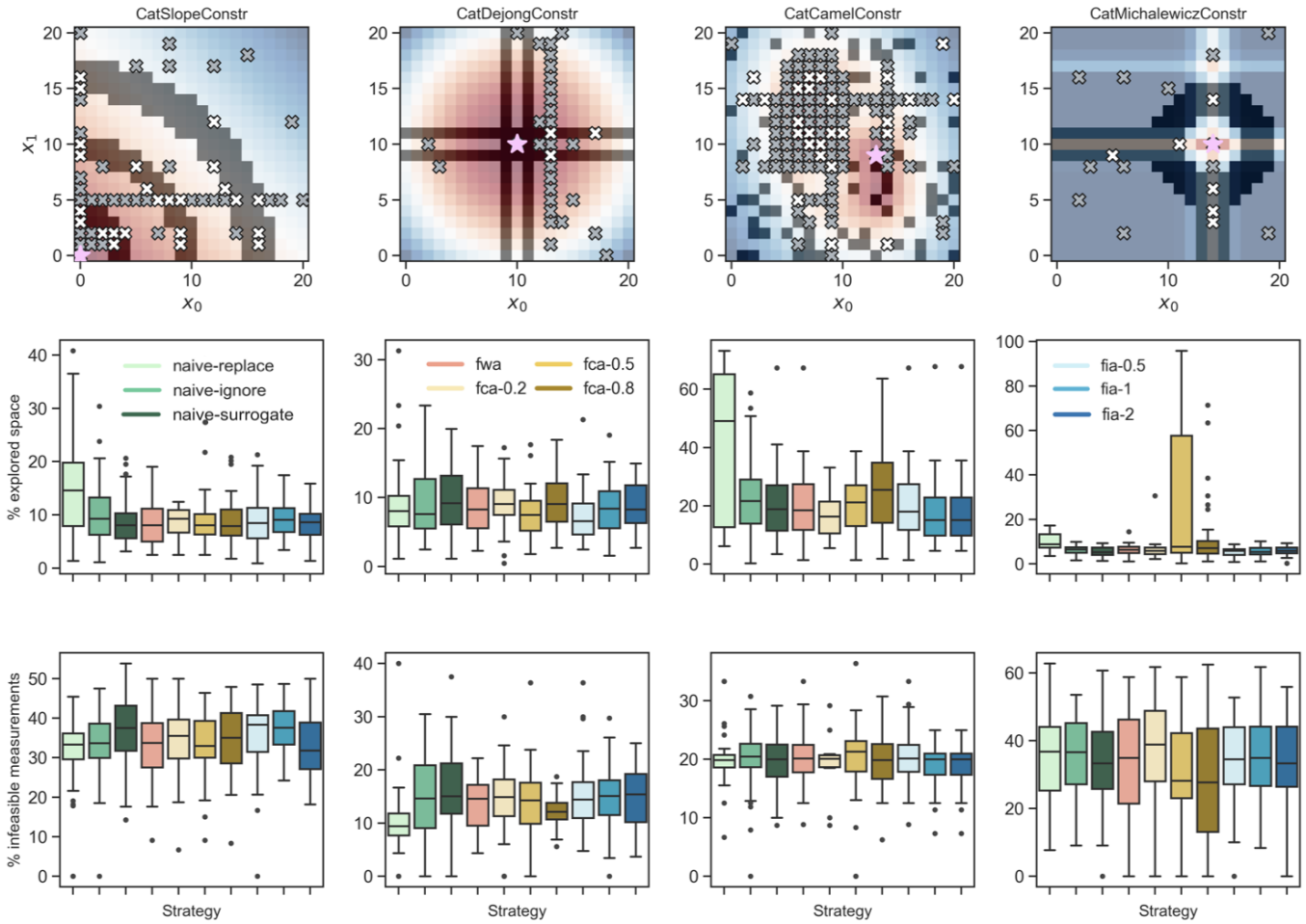

FIG. S5. Constrained optimization benchmarks on analytical functions with categorical, unordered parameters. The upper row shows heatmaps of the discrete optimization domain. Gray crosses show feasible measurements, while white crosses show infeasible measurements, and pink stars show the location of the unconstrained global optima. The middle row shows box-and-whisker plots of the distribution of the percentage of explored parameter space needed for each strategy to identify the global optimum. Each strategy was run 100 independent times. The bottom row shows box-and-whisker plots of the distribution of the number of infeasible measurements queried by each strategy.

C. Details of the (poly)pharmacological design of BCR-Abl fusion protein inhibitors application

1. *Extrapolation of kinase dataset using supervised learning*

Desai *et al.*[7] report a flow technology platform applied to discovering novel Abl kinase inhibitors, in which the flow system enables synthesis, purification and analysis via an on-line bioassay. The authors conduct a search for kinase inhibitors by combining 10 templates with 27 hinge binding aromatic alkynes to span a space of 270 molecules joined via a Sonogashira coupling reaction.

The authors subject 96/270 possible structures to synthesis and bioassay analysis. Of these 96, 71 were synthetically successful and had measurements of the IC_{50}. Formally, we have datasets $\mathcal{D}^K_{f,\text{exp}} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{K-L}$ and $\mathcal{D}^K_{c,\text{exp}} = \{\boldsymbol{x}_i, \tilde{y}_i\}_{i=1}^{K}$, where $K = 96$ and $L = 96 - 71 = 25$. The subscript *exp* is used to indicate that this data comes purely from experiment. We use supervised learning to extend the datasets $\mathcal{D}^K_{f,\text{exp}}$ and $\mathcal{D}^K_{c,\text{exp}}$ to include measurements for all 270 candidate inhibitors. We first train a classifier, which is trained on the data in $\mathcal{D}^K_{c,\text{exp}}$ and makes predictions binary predictions on the remaining $270 - 96 = 174$ candidates. The NGBoost algorithm is used for supervised learning. NGBoost is a flexible, scalable approach which generalizes gradient boosting to probabilistic regression.[8] Extended-connectivity fingerprints (ECFPs)[9] with a radius of 3 and dimension of 1024 represent the inhibitor's molecular structures, *i.e.* $\boldsymbol{x}_i \in \{0,1\}^{1024}$. This produces the hybrid experimental-synthetic feasibility dataset $\mathcal{D}^K_{c,\text{hyb}}$ which is ultimately used as the feasibility oracle during our optimization experiments, where $K = 270$.

Next, a regressor is trained on the experimental data to extrapolate IC_{50} values for the entire inhibitor candidate space. Here, we again use NGBoost and the identical ECFP molecular representation. This time, the model is trained on $\mathcal{D}^{K}_{f,\mathrm{exp}}$, and makes predictions *only* for the structures which that were deemed to have feasible syntheses by our classifier. The results of the NGBoost classifier and regressor are shown in Figure S6 and S7 respectively. Statistics are reported as averages over a 5-fold cross-validation procedure. The average test set AUC-ROC value for our classifier is 0.75, indicating acceptable discriminative performance. The regressor achieves an average test set Pearson coefficient of 0.94, indicating excellent performance.

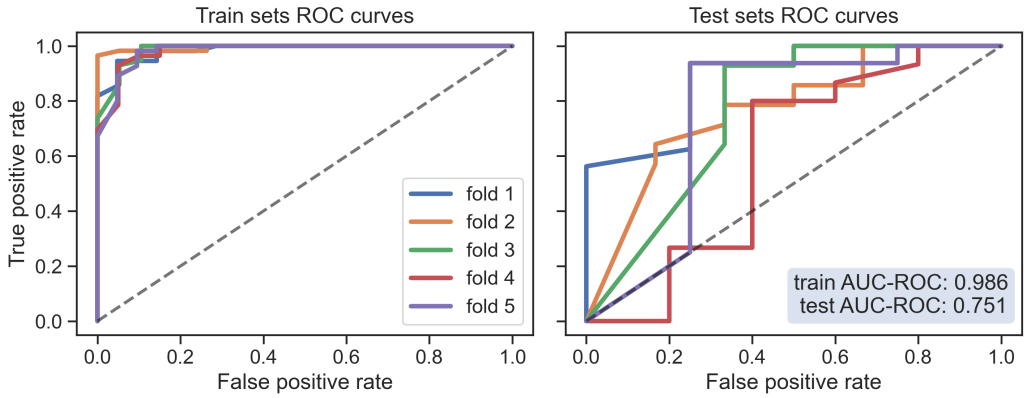

FIG. S6. ROC curves and associated average AUC-ROC values of the NGBoost model predicting Abl kinase inhibitor synthesizability (binary classification) on cross-validation folds of the dataset reported by Desai *et al.*[7]

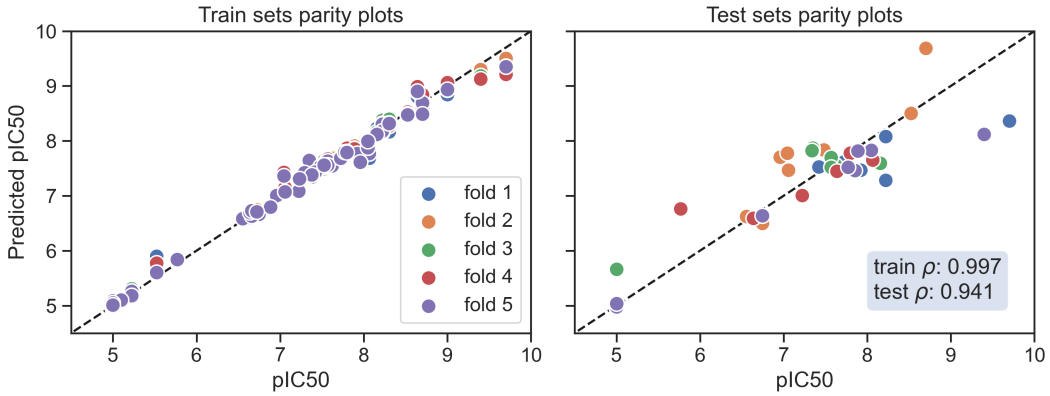

FIG. S7. Parity plots and associated average Pearson coefficient values of the NGBoost model predicting Abl kinase inhibitor pIC_{50} (regression) on cross-validation folds of the dataset reported by Desai *et al.*[7] The pIC_{50} is the negative base 10 logarithm of the IC_{50}.

2. *Polypharmacology of BCR-Abl fusion protein inhibitors*

Imatinib is a type of cancer growth blocker and an example of a tyrosine kinase inhibitor.[10,11] It was originally designed as a selective inhibitor against the BCR-Abl fusion protein, but was subsequently shown to inhibit the non-oncogenic C-Abl kinase in normal cells. This additional activity was reported to be relevant for the drug's efficacy against chronic myeloid leukemia, with the drug's polypharmacological profile potentially forming the basis for its therapeutic activity.[12]

3. Simulation of additional objectives using MPNN emulators

We create two virtual optimization objectives by emulating the log half maximal inhibitory concentration (pIC_{50}) of the inhibitors reported by Desai *et al.* for two additional tyrosine kinase receptors: the platelet-derived growth factor receptor, and stem cell growth factor KIT. Briefly, this will constitute a multi-objective optimization problem with the same unknown constraints as in the single objective case in which ATLAS is tasked with finding a single inhibitor that has a polypharmacological effect, that is, a pharmaceutical agent that can act on multiple targets or disease pathways.

Datasets of pIC_{50} values for tyrosine kinase inhibitor drugs against both KIT and PDGF were procured from BindingDB.[13–16] The KIT dataset consisted of 1810 unique drug structures, while the PDGF dataset consisted of 2242 structures. On each dataset, a message-passing neural network (MPNN) is trained. We use the CHEMPROP[17–19] framework to train the MPNNs. A subset of the hyperparameters of each network (hidden size, depth, dropout and number of feed-forward layers) were tuned using 50 iterations of Bayesian optimization as implemented in the HYPEROPT package.[20] The resulting parity plots for each architecture are shown in Fig. S8. The models achieve Pearson coefficients ≥ 0.88 on the train and test sets (80/10/10 splits), indicating satisfactory performance for our emulator networks.

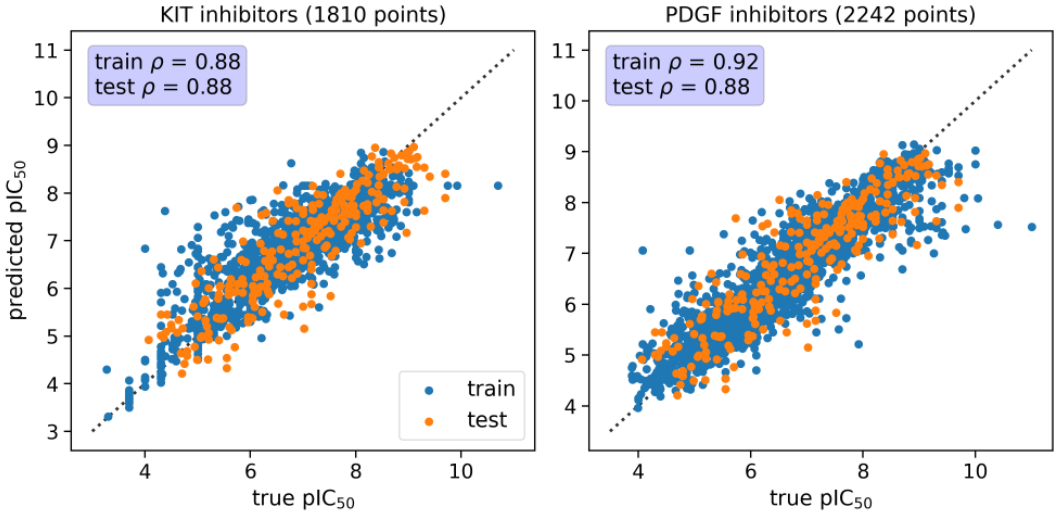

FIG. S8. Parity plots and associated Pearson coefficients for MPNN models trained to reproduce pIC_{50} values on the KIT and PDGF datsets.

The trained MPNN models are then asked to predict pIC_{50} values for the inhibitors in the chemical space defined by Desai *et al.*[7]. KDE plots in Fig. S9 show the distributions of pIC_{50} values for each of the tyrosine kinase receptors across the 270 inhibitors. Importantly, we do not claim that these predictions of inhibitory concentration are accurate with respect to experimental determinations. We are only concerned with constructing a somewhat realistic optimization problem to benchmark our feasibility-aware acquisition functions in a multi-objective setting.

4. Generation of physicochemical descriptors

To help guide optimization for the kinase inhibitor application, ATLAS uses a physicochemical descriptor representation of the molecular template and alkyne options. We hand-select a set of 17 2d molecular descriptors from the Mordred Python library.[21] The descriptors are as follows: molecular weight (`MW`), topological radius (`Radius`), topological diameter (`Diameter`), topological polar surface area (`TopoPSA`), rotable bonds ratio (`RotRatio`), number of rotable bonds (`nRot`), bond polarizability (`bpol`), atomic polarizability (`apol`), number of hydrogen bond donor (`nHBDon`), hybridization ratio (`HybRatio`), Wildman-Crippen LogP (`SLogP`), number of hydrogen bond donor (`nHBDon`), number of SP3 carbon bound to 1 other carbon (`C1SP3`), number of SP2 carbon bound to 3 other carbons (`C3SP2`), number of SP2 carbon bound to 2 other carbons (`C2SP2`), number of SP2 carbon bound to 1 other carbons (`C3SP2`), number of heteroatoms (`nHetero`), and the number of heavy atoms (`nHeavyAtom`). The same set of descriptors are used for both template and alkyne molecules, thus the descriptor vector $\boldsymbol{x} \in \mathbb{R}^{34}$ is utilized by the optimizer.

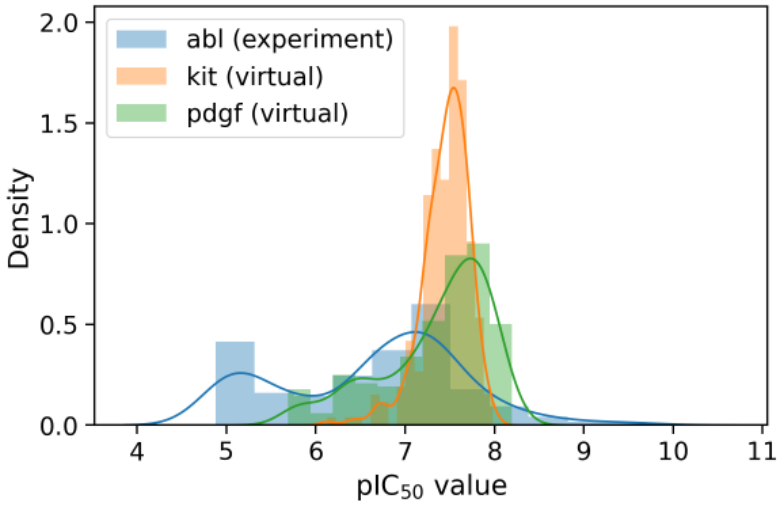

FIG. S9. KDE plots showcasing the distributions of pIC_{50} values for each of the 3 objectives.

D. Details of the hybrid organic-inorganic halide perovskite design application

Körbel *et al.* construct a search space of hybrid organic-inorganic perovskite (HOIP) materials accessed by chemical substitution of the form $A^+B^{2+}X_3^-$ (main text Fig. 4a).[22] The goal of the screening was to find thermodynamically stable HOIPs with small effective masses and optimal band gaps for photovoltaics. The search space consisted of 11 molecular cations (A), 29 divalent metals (B), and 4 halogens (X) for a total of 1276 possible HOIPs.

1. *Geometric and electronic descriptors of HOIP components*

To help guide optimization for the HOIP application, we represent the HOIP component options with vector-valued descriptors, composed of geometric and electronic properties. Namely, the inorganic options are characterized by their mass, electronegativity, electron affinity, and ionization energy. The organic compounds were characterized by their molecular weight, radius of gyration, atomization energy, dipole moment norm, and frontier molecular orbital energies. Molecular conformers are generated using OpenBabel[23] and optimized using the xTB-GFN2 semi-empirical Hamiltonian.[24] Electronic descriptors were calculated at the HSEH1PBE/Def2QZVPP level of theory on a SuperFineGrid with the Gaussian program package.[25] Geometric descriptors were calculated with the Mordred Python package.[21] The radius of gyration was computed for the lowest energy conformer. In sum, a 14-element descriptor vector is processed by the optimizer for this application, $\boldsymbol{x} \in \mathbb{R}^{14}$.

[1] Kraft D, et al. A software package for sequential quadratic programming. Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt fur Luft- und Raumfahrt. 1988.

[2] Balandat M, Karrer B, Jiang DR, Daulton S, Letham B, Wilson AG, et al. BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In: Advances in Neural Information Processing Systems 33; 2020. Available from: http://arxiv.org/abs/1910.06403.

[3] Blank J, Deb K. pymoo: Multi-Objective Optimization in Python. IEEE Access. 2020;8:89497-509.

[4] Fortin FA, De Rainville FM, Gardner MA, Parizeau M, Gagné C. DEAP: Evolutionary Algorithms Made Easy. Journal of Machine Learning Research. 2012;13:2171-5.

[5] De Rainville FM, Fortin FA, Gardner MA, Parizeau M, Gagné C. Deap: A python framework for evolutionary algorithms. In: Proceedings of the 14th annual conference companion on Genetic and evolutionary computation; 2012. p. 85-92.

[6] Hickman RJ, Aldeghi M, Häse F, Aspuru-Guzik A. Bayesian optimization with known experimental and design constraints for chemistry applications. Digital Discovery. 2022 Sep. Publisher: RSC. Available from: https://pubs.rsc.org/en/content/articlelanding/2022/dd/d2dd00028h.

[7] Desai B, Dixon K, Farrant E, Feng Q, Gibson KR, van Hoorn WP, et al. Rapid Discovery of a Novel Series of Abl Kinase Inhibitors by Application of an Integrated Microfluidic Synthesis and Screening Platform. Journal of Medicinal Chemistry. 2013 Apr;56(7):3033-47. Publisher: American Chemical Society.

[8] Duan T, Avati A, Ding DY, Basu S, Ng AY, Schuler A. NGBoost: Natural Gradient Boosting for Probabilistic Prediction. CoRR. 2019;abs/1910.03225. Available from: http://arxiv.org/abs/1910.03225.

[9] Rogers D, Hahn M. Extended-Connectivity Fingerprints. Journal of Chemical Information and Modeling. 2010;50(5):742-54. PMID: 20426451. Available from: https://doi.org/10.1021/ci100050t.

[10] Deininger MWN, Druker BJ. Specific targeted therapy of chronic myelogenous leukemia with imatinib. Pharmacological Reviews. 2003 Sep;55(3):401-23.

[11] Iqbal N, Iqbal N. Imatinib: A Breakthrough of Targeted Therapy in Cancer. Chemotherapy Research and Practice. 2014;2014:357027.

[12] Cohen P, Cross D, Jänne PA. Kinase drug discovery 20 years after imatinib: progress and future directions. Nature Reviews Drug Discovery. 2021 Jul;20(7):551-69.

[13] Chen X, Liu M, Gilson MK. BindingDB: a web-accessible molecular recognition database. Combinatorial Chemistry & High Throughput Screening. 2001 Dec;4(8):719-25.

[14] Chen X, Lin Y, Liu M, Gilson MK. The Binding Database: data management and interface design. Bioinformatics (Oxford, England). 2002 Jan;18(1):130-9.

[15] Liu T, Lin Y, Wen X, Jorissen RN, Gilson MK. BindingDB: a web-accessible database of experimentally determined protein–ligand binding affinities. Nucleic Acids Research. 2007 Jan;35(Database issue):D198-201. Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1751547/.

[16] Gilson MK, Liu T, Baitaluk M, Nicola G, Hwang L, Chong J. BindingDB in 2015: A public database for medicinal chemistry, computational chemistry and systems pharmacology. Nucleic Acids Research. 2016 Jan;44(D1):D1045-53.

[17] Yang K, Swanson K, Jin W, Coley C, Eiden P, Gao H, et al. Analyzing Learned Molecular Representations for Property Prediction. Journal of Chemical Information and Modeling. 2019 Aug;59(8):3370-88. Publisher: American Chemical Society.

[18] Stokes JM, Yang K, Swanson K, Jin W, Cubillos-Ruiz A, Donghia NM, et al. A Deep Learning Approach to Antibiotic Discovery. Cell. 2020 Feb;180(4):688-702.e13. Publisher: Elsevier.

[19] Heid E, Green WH. Machine Learning of Reaction Properties via Learned Representations of the Condensed Graph of Reaction. Journal of Chemical Information and Modeling. 2022 May;62(9):2101-10. Publisher: American Chemical Society.

[20] Bergstra J, Komer B, Eliasmith C, Yamins D, Cox DD. Hyperopt: a Python library for model selection and hyperparameter optimization. Computational Science & Discovery. 2015;8(1):014008. Available from: http://stacks.iop.org/1749-4699/8/i=1/a=014008.

[21] Moriwaki H, Tian YS, Kawashita N, Takagi T. Mordred: a molecular descriptor calculator. Journal of Cheminformatics. 2018 Feb;10(1):4. Available from: https://doi.org/10.1186/s13321-018-0258-y.

[22] Körbel S, Marques MAL, Botti S. Stable hybrid organic–inorganic halide perovskites for photovoltaics from ab initio high-throughput calculations. Journal of Materials Chemistry A. 2018 Apr;6(15):6463-75. Publisher: The Royal Society of Chemistry. Available from: https://pubs.rsc.org/en/content/articlelanding/2018/ta/c7ta08992a.

[23] O'Boyle NM, Banck M, James CA, Morley C, Vandermeersch T, Hutchison GR. Open Babel: An open chemical toolbox. Journal of Cheminformatics. 2011 Oct;3(1):33. Available from: https://doi.org/10.1186/1758-2946-3-33.

[24] Bannwarth C, Caldeweyher E, Ehlert S, Hansen A, Pracht P, Seibert J, et al. Extended tight-binding quantum chemistry methods. WIREs Computational Molecular Science. 2021;11(2):e1493. Available from: https://onlinelibrary.wiley.com/doi/abs/10.1002/wcms.1493.

[25] Hensman J, Fusi N, Lawrence ND. Gaussian Processes for Big Data. arXiv; 2013. Available from: https://arxiv.org/abs/1309.6835.